Exercise 2

Time tip: Roughly 45sec to 1min per 1pt
Q1) [3pts] Which of the following is/are (a) major reason(s) for the rising adoption of cloud computing?

A  Pay-as-you-go economics
B  Manageability
C  Data privacy
D  Both A & B
E  All of A, B, & C
Q2) [3 x 3pts] Suppose an SQL query takes 20min to run on a single worker node and \(x\) min when run on 5 worker nodes. What is the speedup for the given value of \(x\)? Is the speedup linear, sublinear, or superlinear?

A. \(x = 7\) min

B. \(x = 4\) min

C. \(x = 3\) min

A. Speedup = \(20/7 = 2.86x\); < 5x, sublinear

B. Speedup = \(20/4 = 5x\); = 5x, linear

C. Speedup = \(20/3 = 6.67x\); > 5x, superlinear
Q3) [3 x 3pts] Suppose an ML training workload takes 40min to run on a single worker node. We then triple the dataset size, say, to help improve accuracy and use 3 worker nodes. It now takes $x$ min. What is the speedup for the given value of $x$? Is the scaleup linear, sublinear, or superlinear?

A. $x = 50$min  
   A. Speedup = $40/50 = 0.8x$; < 1x, sublinear

B. $x = 40$min  
   B. Speedup = $40/40 = 1x$; = 1x, linear

C. $x = 35$min  
   C. Speedup = $40/35 = 1.14x$; > 1x, superlinear
Q4) [4pts] What is the speedup yielding by this task-parallel schedule on 3 workers against 1 worker?

3-worker time = 30 (from chart)
1-worker time =
10+25+20+10+5+5 = 75
Speedup = 75 / 30 = 2.5x
Q5) [5pts] Consider the same Gantt Chart as in Q4. Suppose you are given that T6 and T5 depend on T2. What will be the new speedup with 3 more workers for task-parallel execution? Explain your answer succinctly.

Speedup will still be only 2.5x.
T1, T2, and T3 can start in parallel; T4 can run after T1 on that same worker; but T5 and T6 can not start anywhere until T2 finishes. But once T2 finishes, T5 and T6 can run in parallel. So, Gantt chart even with 6 workers ends up looking the same/similar, with same completion time of 30.
Exercise

Q6) Consider the following task graph with the task lengths shown. You are given 3 workers to execute this graph in a task-parallel manner like discussed in class.

A. [3pts] What is the lowest possible completion time of this workload?

**Answer:**

25; it is the longest path = 15 + 5 + 5 = 25

B. [5pts] What is the highest possible speedup of this workload on 3 workers vs its runtime on just 1 worker without idling?

**Answer:**

2.4x; total 1-worker time = 5+15+20+10+5+5 = 60
Regardless of # workers, lowest possible completion time is 25 as above.
So, highest possible speedup = 60/25 = 2.4x

C. [5pts] What is the total idle time across all workers in a schedule that yields the highest speedup?

**Answer:**

15; with 3 workers, we can hit above max speedup, with completion time of 25.
Total time consumed by 3 workers = 3 x 25 = 75
So, total idle time = 75 - 60 = 15
Exercise

Q7) Consider the following task graph with some task lengths shown and some unknown/hidden.

A. [2pts] What is the degree of parallelism in this task graph?
   2; only 2 tasks can run concurrently based on given graph

B. [4pts] You are given that the lowest possible completion time of this workload is 30. What are the maximum possible values of x and y?
   y = 10, x = 20. Lowest possible completion time is longest path, i.e., max (20+y, 20+10, x+10). Set it to 30. So, max value of y = 30-10 = 20; likewise, max value of x = 30-10 = 20.

C. (Advanced; Optional) [8pts] Instead of B, you are now given that y = 2x. What is the highest possible speedup for this workload with task-parallel execution?
   1.5x; lowest possible completion time now is max (20+2x, 20+10, x+10). To get highest possible speedup, you need its lowest possible value, which is just 30. That means 20+2x can be max 30, which means x can be max 5. 1-worker time = x+y+30 = 3*(x+10). So, max speedup is 3*(x+10)/30. Setting x=5 then gives us 3*15/30 = 1.5x
Q8) [3 x 3pts] Suppose an ML training program has 2 parts: a parallelizable part that takes 1hr when run with 1 machine and an inherently sequential part that takes 20min on 1 machine. Suppose the parallelizable part yields perfect linear speedup on x workers. What is the overall speedup of the training when run on n workers? Actual runtime?

A. n = 2  
B. n = 6  
C. n -> infinity

A. (60+20) / (30+20) = 1.6x  
B. (60+20) / (10+20) = 2.67x  
C. (60+20) / (0+20) = 4x
(Advanced; Optional) Q9) [8pts] You are given a workload with \( n \) tasks, each of length \( k \) units. You are allowed to use task parallelism as discussed in class. What is the lowest possible completion time of this workload? What should the task graph look like for that and how many workers are necessary to achieve that completion time?

Lowest compl. time is \( k \). Every task must run in parallel. So, the task graph simply has all tasks independent of each other. Exactly \( n \) workers are necessary, since there are \( n \) tasks. Any less will raise compl. time. Any more will just sit idle.